Why You Should Write dApps in Go

June 4th, 2025
Belgrade, Serbia

Agenda

- Step 1: What is Go
- Step 2: Write app in Go
- Step 3: ???
- Step 4: ~~Profit~~ Have a complete dApp

What is Go, and why Go?

Go is a general-purpose,
compiled language…

…which has been
around since ~2009.

It’s statically typed, and it’s kind of like the “C of the 21st century”.

Why Go?

Used to build backend services, protocols which require efficiency…

Has a great standard library…

and an amazing set of tools to work with…

But most importantly:
it's easy to learn

It has a straightforward, easy learning curve

and has lots of battle-tested libraries and resources
by a large dev community.

Let’s (try) Go!

So, what if we had:

- Automatic persistence
- Language-based app features:
  - Exported functions & methods = APIs
  - Top-level vars = DB/state
- Built-in security and decentralization
- Easy to make frontend
- One CLI command to publish our app

Well, we do - with Gno

Gno

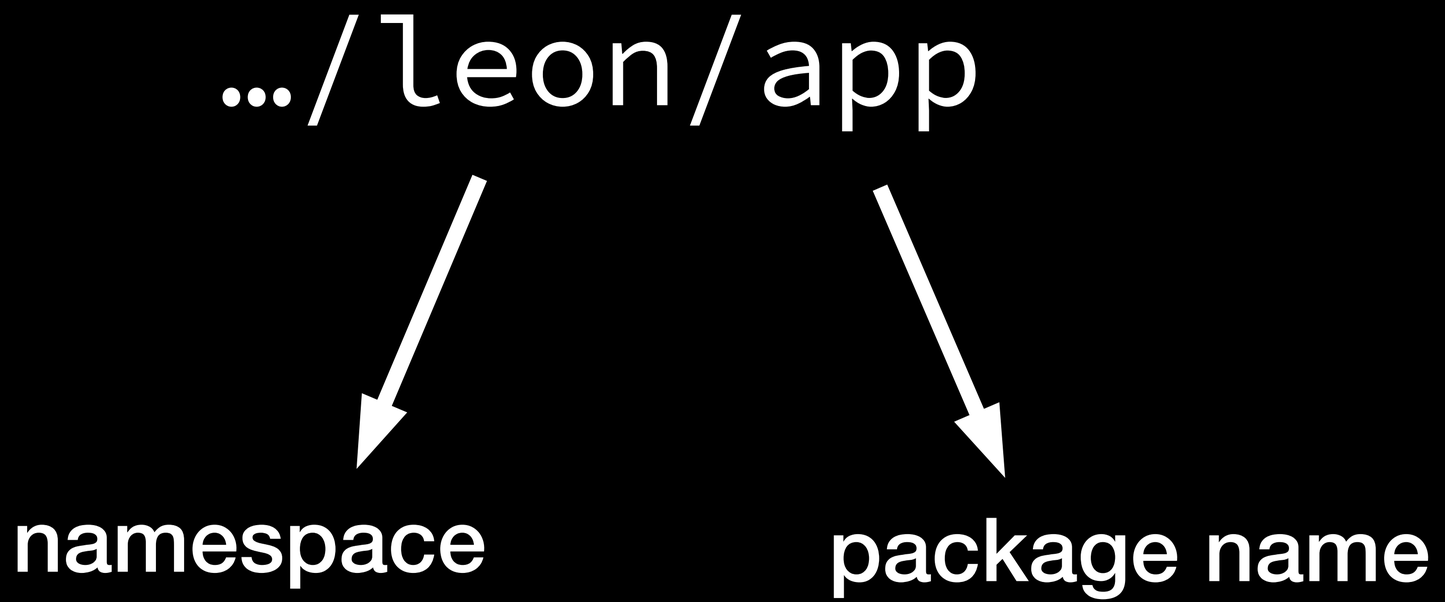
```
package alice

var x int

func GetX() int {
        return x
}

func SetX(n int) {
        x = n
}
```

```
package bob

import "alice"

func IncrAlice() {
        x := alice.GetX()
        alice.SetX(x+1)
}
```

More on Gno…

- Modeled after Go 1.18
- Interpreted instead of compiled
- Currently does not support all go features, such as generics & goroutines
- All apps are ID’d by something called a “pkgpath”

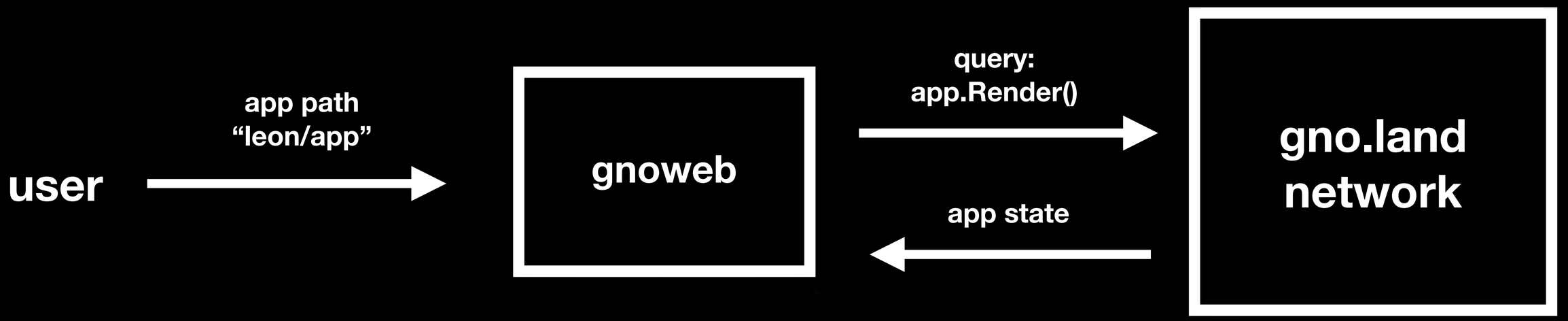
Gno Package Path



gnoweb

- Universal web frontend for Gno.land, which anyone can run
- Uses a pre-built query to get app state, to display it to the user
- … which is why there’s no need for JS/TS - just write Gno

gnoweb


user
app path
“leon/app”
gnoweb
query:
app.Render()
app state
gno.land
network

so… where’s the dApp in Go?

Tooling & Ecosystem

gnokey

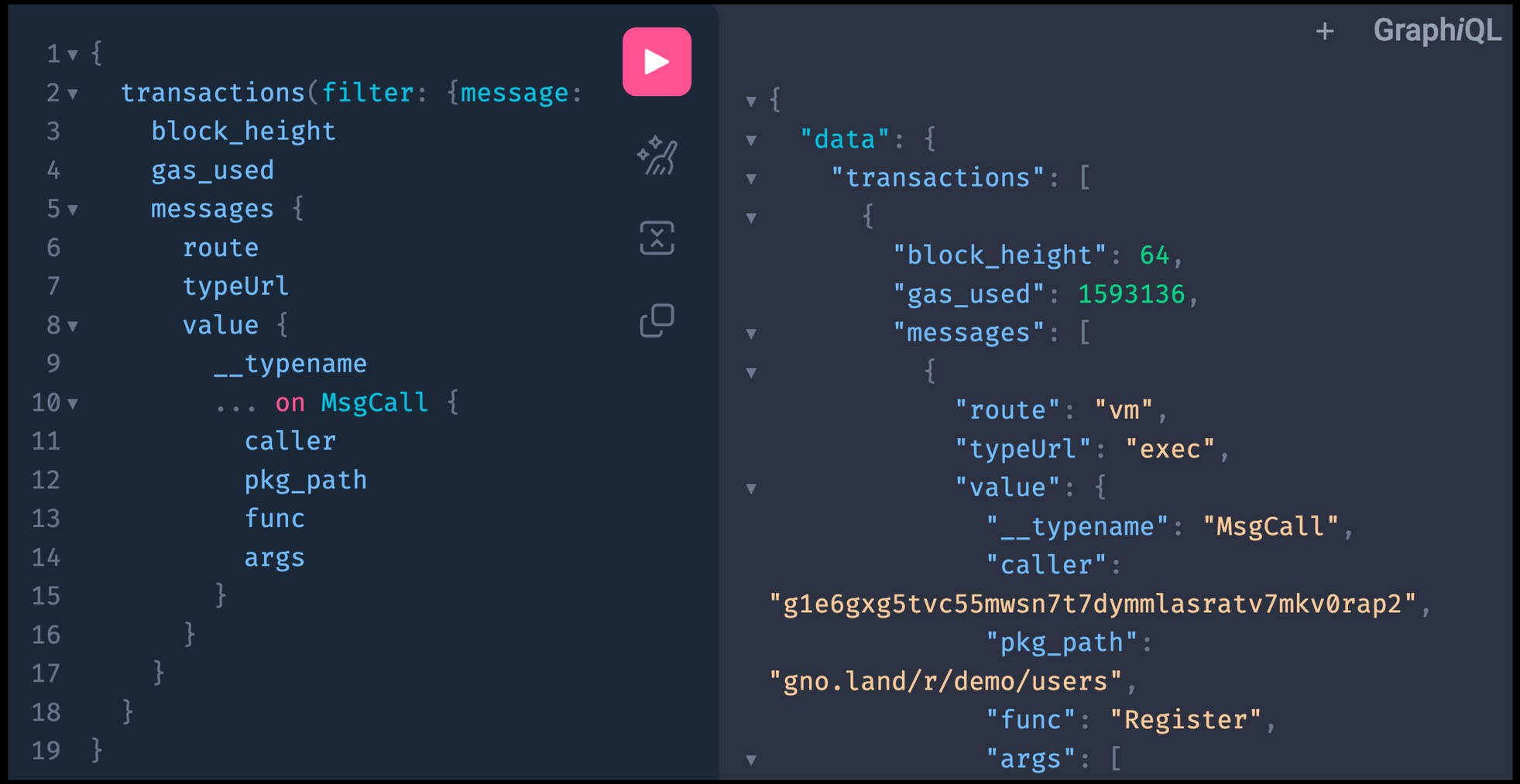
```
❯ gnokey --help
USAGE
  <subcommand> [flags] [<arg>...]

gno.land keychain & client

SUBCOMMANDS
  add         adds key to the keybase
  delete      deletes a key from the keybase
  generate    generates a bip39 mnemonic
  export      exports private key armor
  import      imports encrypted private key armor
  list        lists all keys in the keybase
  sign        signs the given tx document and saves it to disk
  verify      verifies the document signature
  query       makes an ABCI query
  broadcast   broadcasts a signed document
  maketx      composes a tx document to sign
```

gnodev

- Local development solution stack for Gno
- Has an local in-memory blockchain node, gnoweb, account premining & a directory watcher with automatic code reloading

```
❯ gnodev
Accounts      | I default address imported name=test1 addr=g1jg8mtutu
9khhfwc4nxmuhcpftf0pajdhfvsqf5
              | I pkgs loaded path="[{/Users/sasurai/Desktop/gno/gno/
examples g1jg8mtutu9khhfwc4nxmuhcpftf0pajdhfvsqf5 }]"
Node          | I node started lisn=tcp://127.0.0.1:26657 chainID=dev
GnoWeb        | I gnoweb started lisn=http://127.0.0.1:8888
--- READY     | I for commands and help, press `h`
```

tx-indexer


{
transactions(filter: {message:
block_height
gas_used
messages {
route
typeUrl
value {
__typename
... on MsgCall {
caller
pkg_path
func
args
}
}
}
}
}
GraphiQL
{
"data": {
"transactions": [
{
"block_height": 64,
"gas_used": 1593136,
"messages": [
{
"route": "vm",
"typeUrl": "exec",
"value": {
"__typename": "MsgCall",
"caller":
"g1e6gxg5tvc55mwsn7t7dymmlasratv7mkv0rap2",
"pkg_path":
"gno.land/r/demo/users",
"func": "Register",
"args": [

Gno Playground

play.gno.land

Ecosystem

adena

gnoscan

Gno Native Kit

Flippando

GnoSwap

GNO JS/TS Client

TERITORI

Gno.land Go Client

gno.studio

Tendermint2 JS/TS Client

dSocial

gnokey mobile

GnoChess BETA

Chess: The Gnolang Way

Opportunities

- Be among the first (read- early) to build useful dApps, libraries and work on protocol level problems and make an impact on the future of gno.land.
- If you’re looking to start a local/regional community for gno.land, reach out to us!

Grants, Student Programs, and more!

Contact

- Leon Hudak
- DevRel Engineer @ gno.land
- ~4 yrs in Web3
- X, GitHub: @leohhhn

Please take 2 minutes to provide feedback about today’s presentation!